CSAMA 2016: Clustering, classification, and regression with genomic examples

Vince Carey

July 13, 2016

Contents

0.1 Road map

- use cases

- user interface concepts
- cluster analysis components
  - primitive sensitivity analysis
- classifier components
  - role of metapackages like caret/mlr/MLInterfaces

0.2 Use case 1: transcript profiles to distinguish tissue source

- illumina bodymap in GEO
- another application: adequacy of mouse models of human biology

0.3 Species and organ of origin: microarrays and orthologues (McCall et al., *NAR* 2012)

D1014 *Nucleic Acids Research, 2011, Vol. 39, Database issue*

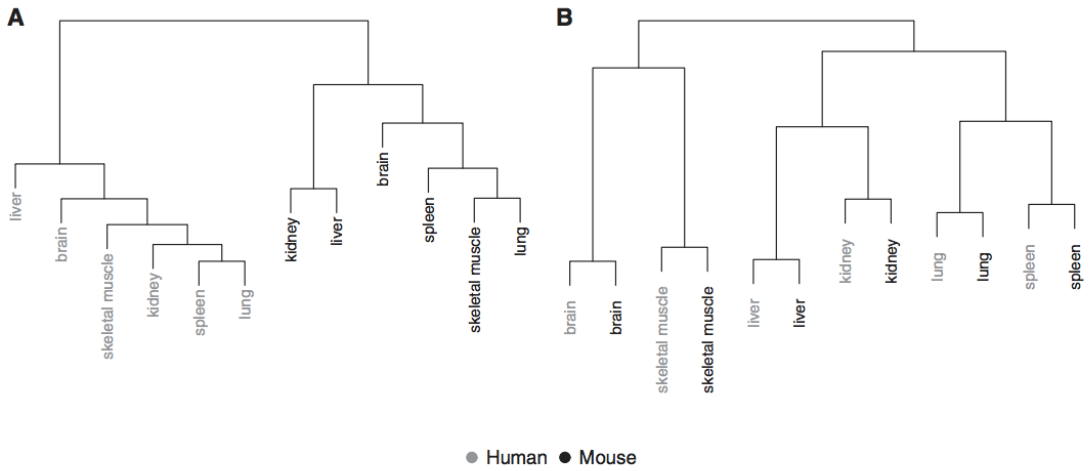
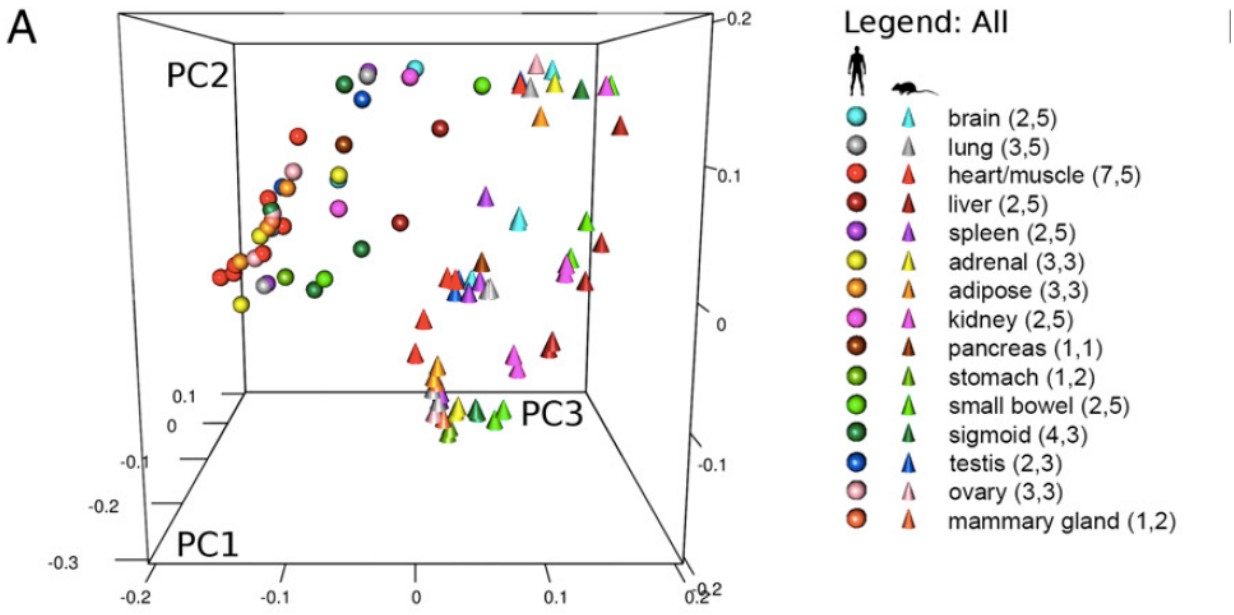


Figure 2. Hierarchical clustering of human and mouse tissue samples using orthologous genes. These are based on (**A**) average expression microarray measurements and (**B**) tissue specific transcriptomes based on averaged barcodes. The same genes were used in (**A**) and (**B**).

0.4 Species, organ of origin, and batch: RNA-seq and orthologues (Lin et al., *PNAS* 2014)



- Between-species disparity stronger than within-organ similarity

0.5 Question

- Distinguishing organ of origin through gene expression patterns
  - McCall *et al.*, *NAR* 2011
  - adjusted arrays yield 85 22215-vectors
  - barcode transformation: transcriptomes cluster by organ
- Comparison of human and mouse transcriptomes
  - Lin *et al.*, *PNAS* 2014
  - mRNA abundance for orthologous genes by RNA-seq, 30 15106-vectors
  - transcriptomes cluster by species

Which one is right?

0.6 Use case 2: Oncotype DX gene signature for breast cancer survival

- 21 genes useful for prediction of breast cancer recurrence
- Paik, Shak, Tang et al. NEJM 2004
- *genefu* package includes notation for the signature (`sig.oncotypedx`)
- We'll consider the capacity of the gene set for predicting overall survival in a classic breast cancer dataset (van de Vijver 2002) as packaged in `genefu`

0.7 Setup for NKI breast cancer expression/clinical data

```
library(genefu); library(survival)
data(nkis)
map = as.character(annot.nkis$NCBI.gene.symbol)
names(map) = as.character(annot.nkis$probe)
ndata.nkis = data.nkis
colnames(ndata.nkis) = map[colnames(data.nkis)]
cbind(ndata.nkis[1:4,1:4], demo.nkis[1:4,5:8])
##           ESR1 TBC1D9  GATA3   CA12 grade node size age
## NKI_123  0.195 -0.114  0.202  0.158     3    0  2.0  48
## NKI_327  0.034  0.033  0.158  0.103     2    1  2.0  49
## NKI_291 -0.417  0.140  0.006 -0.266     2    1  1.2  39
## NKI_370  0.429  0.352 -0.050  0.236     1    1  1.8  51
```

0.8 Label expression columns with appropriate symbols; test

```
nkSurv = Surv(demo.nkis$t.os, demo.nkis$e.os)
odata = ndata.nkis[, intersect(as.character(sig.oncotypedx$symbol),
    colnames(ndata.nkis))]
fullnk = cbind(demo.nkis, odata)
coxph(nkSurv~er+age, data=fullnk)
## Call:
## coxph(formula = nkSurv ~ er + age, data = fullnk)
##
##        coef exp(coef) se(coef)     z      p
## er  -1.0018    0.3672   0.3425 -2.92 0.0034
## age -0.0328    0.9677   0.0271 -1.21 0.2268
##
## Likelihood ratio test=10.1  on 2 df, p=0.00657
## n= 129, number of events= 36
##    (21 observations deleted due to missingness)
```

0.9 Create a survival tree using all available clinical and expression data

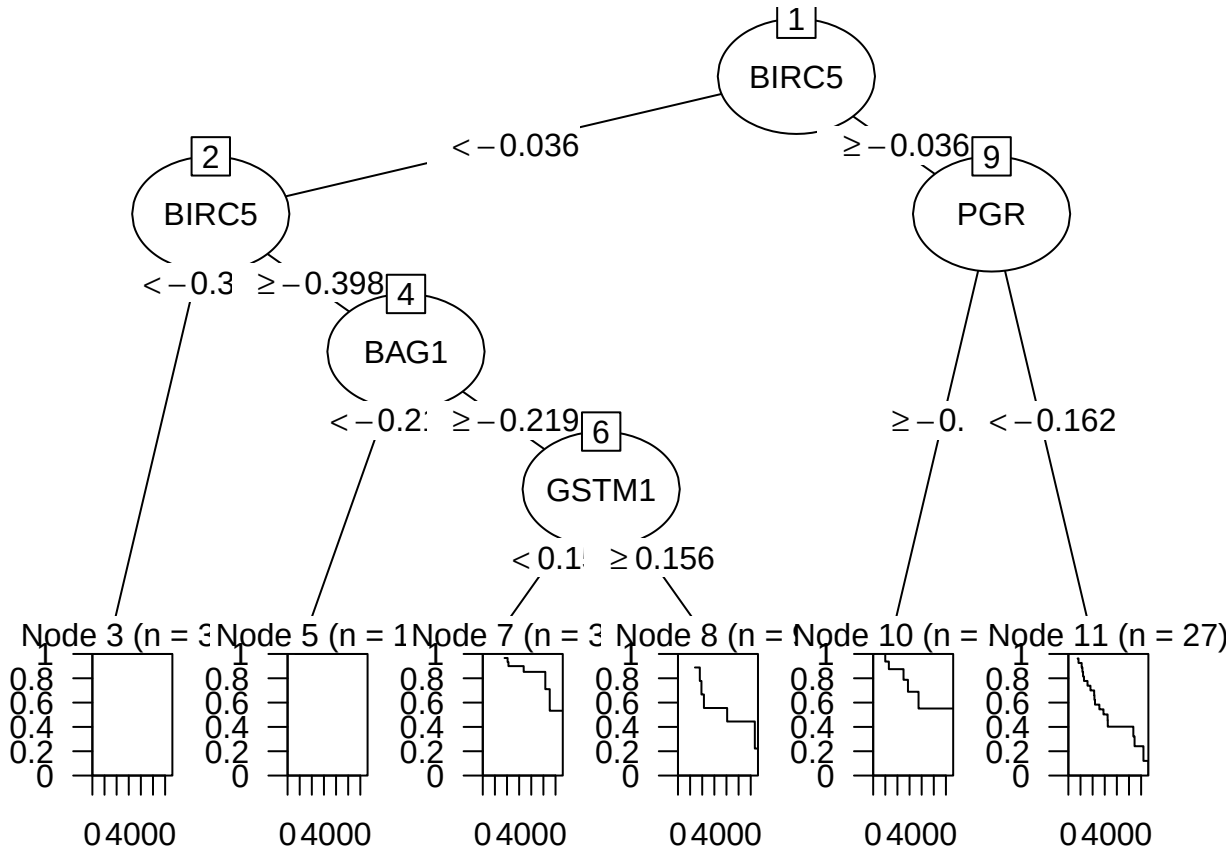
```
rfullnk = fullnk[,-c(1,2,3,9,10,11,12,13,14,17,18,19)]
library(rpart); r1 = rpart(nkSurv~.,data=rfullnk)
r1
## n=129 (21 observations deleted due to missingness)
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 129 146.652400 1.00000000
##    2) BIRC5< -0.0365 85  62.712830 0.47436610
##      4) BIRC5< -0.3975 32   1.801804 0.09909801 *
##      5) BIRC5>=-0.3975 53  52.568420 0.70984040
##       10) BAG1< -0.219 14   1.660224 0.16988820 *
##       11) BAG1>=-0.219 39  44.603630 0.96814410
##         22) GSTM1< 0.1565 30  22.464060 0.58792190
##           44) MKI67>=-0.0655 19   8.070774 0.23294560 *
##           45) MKI67< -0.0655 11   7.582306 1.38868000 *
```

```
##         23) GSTM1>=0.1565 9  12.691410 2.77622500 *
##    3) BIRC5>=-0.0365 44  58.962600 2.35960200
##      6) PGR>=-0.1625 17  16.872130 1.05016300 *
##      7) PGR< -0.1625 27  34.118410 3.40043200
##       14) GSTM1< -0.1235 7   5.180967 1.32643500 *
##       15) GSTM1>=-0.1235 20  23.712420 4.39730500 *
```

CRAN package *partykit* enhances tree support in *rpart* and provides many additional models

```
library(partykit)
p1p = as.party(prune(r1, cp=.05))
```

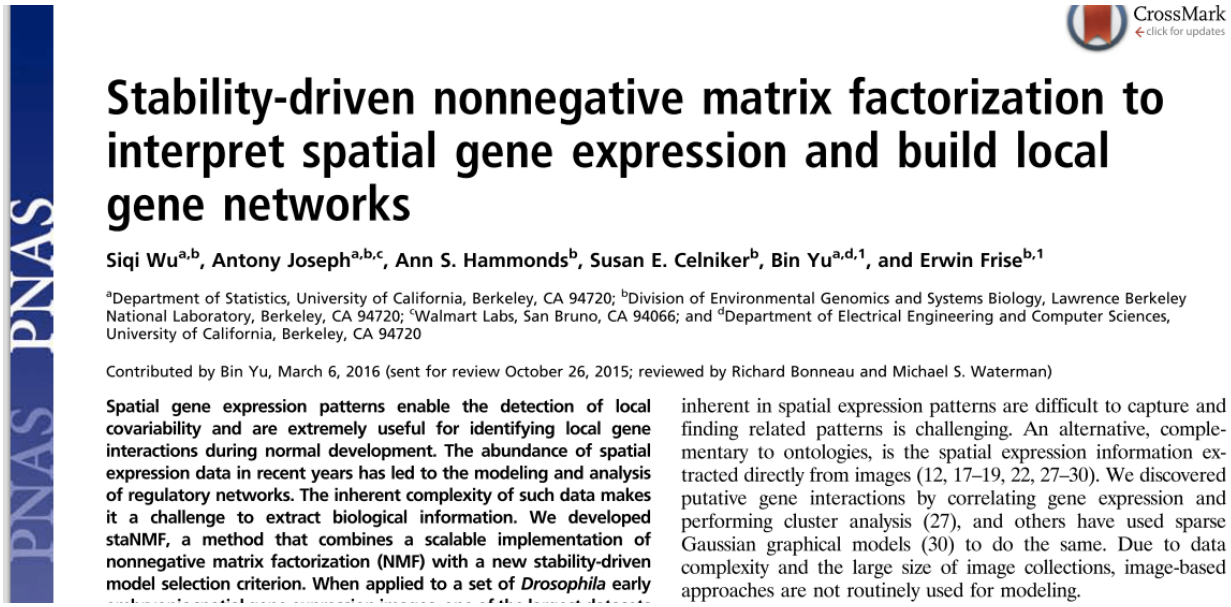
0.10 Visualize the pruned tree along with K-M curves for leaves



0.11 Question

What are the key vulnerabilities of an analysis of this type?

0.12 Use case 3: Cell fate signatures from the fruitfly blastocyst



PNAS

CrossMark click for updates

Stability-driven nonnegative matrix factorization to interpret spatial gene expression and build local gene networks

Siqi Wu[a,b], Antony Joseph[a,b,c], Ann S. Hammonds[b], Susan E. Celniker[b], Bin Yu[a,d,1], and Erwin Frise[b,1]

[a]Department of Statistics, University of California, Berkeley, CA 94720; [b]Division of Environmental Genomics and Systems Biology, Lawrence Berkeley National Laboratory, Berkeley, CA 94720; [c]Walmart Labs, San Bruno, CA 94066; and [d]Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720

Contributed by Bin Yu, March 6, 2016 (sent for review October 26, 2015; reviewed by Richard Bonneau and Michael S. Waterman)

Spatial gene expression patterns enable the detection of local covariability and are extremely useful for identifying local gene interactions during normal development. The abundance of spatial expression data in recent years has led to the modeling and analysis of regulatory networks. The inherent complexity of such data makes it a challenge to extract biological information. We developed staNMF, a method that combines a scalable implementation of nonnegative matrix factorization (NMF) with a new stability-driven model selection criterion. When applied to a set of *Drosophila* early

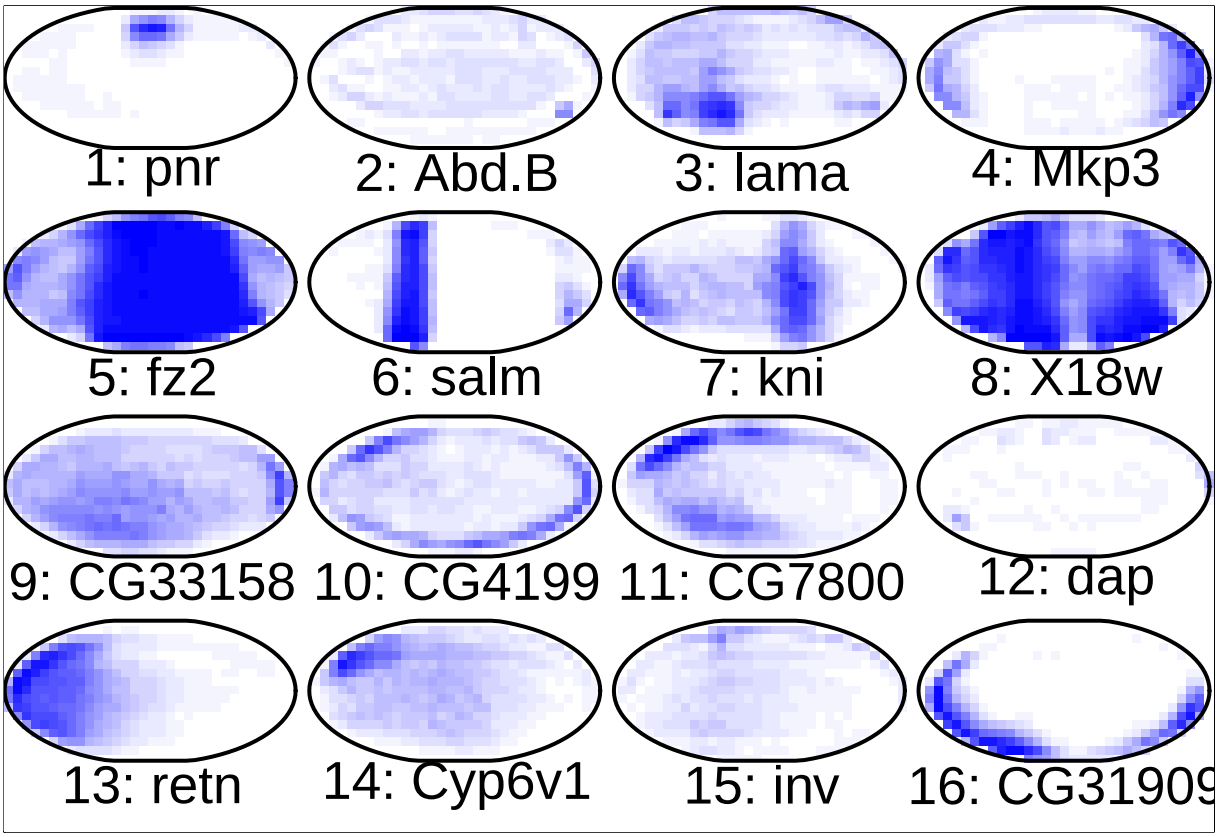
inherent in spatial expression patterns are difficult to capture and finding related patterns is challenging. An alternative, complementary to ontologies, is the spatial expression information extracted directly from images (12, 17–19, 22, 27–30). We discovered putative gene interactions by correlating gene expression and performing cluster analysis (27), and others have used sparse Gaussian graphical models (30) to do the same. Due to data complexity and the large size of image collections, image-based approaches are not routinely used for modeling.

0.13 Data setup

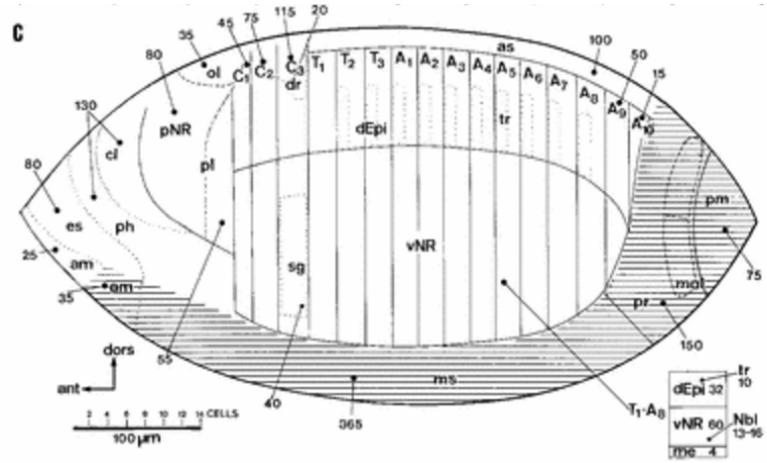
```
library(drosmap) # biocLite("vjcitn/drosmap")
data(expressionPatterns)
data(template); template=template[,-1]
data(uniqueGenes)
uex = expressionPatterns[,uniqueGenes]
uex[1:5,1:5]
##           pnr      Abd.B        lama      Mkp3       fz2
## 1 0.014123479 0.05531271 0.014584370 0.2086337 0.3759253
## 2 0.009015973 0.01234864 0.014212999 0.3222693 0.5585198
## 3 0.023047258 0.01486692 0.013431432 0.3599486 0.5329454
## 4 0.013179102 0.03184486 0.005370888 0.2365888 0.2585371
## 5 0.008820991 0.06811459 0.016528382 0.1136623 0.1034636
```

0.14 Spatial gene-specific patterns

```
imageBatchDisplay(uex[,1:16], nrow=4, ncol=4, template=template)
```


1: pnr
2: Abd.B
3: lama
4: Mkp3
5: fz2
6: salm
7: kni
8: X18w
9: CG33158
10: CG4199
11: CG7800
12: dap
13: retn
14: Cyp6v1
15: inv
16: CG31909

0.15 Can we transform spatial patterns for 701 genes to cohere with this fate map?



of the blastoderm. Notice that the size of the anlagen of the salivary glands (40) and of the dorsal ridge (20) is included in the size of C3. Scales indicate EL% (0–10% and 90%–100% values are distorted due to the reconstruction procedure). *am*: anterior midgut; *as*: amnioserosa; *C3d*: dorsal ridge; *cl*: clypeolabrum; *dEpi*: dorsal epidermis; *dr*: dorsal ridge; *es*: oesophagus; *mp*: Malpighian tubes; *ms*: mesoderm; *ol*: optic lobes; *p*: gnathal protuberances; *ph*: pharynx; *pl*: procephalic lobe; *pm*: posterior midgut; *pNR*: procephalic neurogenic region; *pr*: proctodeum; *sg*: salivary gland; *tr*: tracheae; *vNR*: ventral neurogenic region; *C1–C3*: gnathal segments; *C1p*: mandible; *C2p*: maxilla; *C3p*: labium; *T1–T3*: thoracic segments; *A1–A10*: abdominal segments. See text for further details

0.16 Idea: NMF (Brunet, Tamayo, Golub, Mesirov PNAS 2004) for clustering

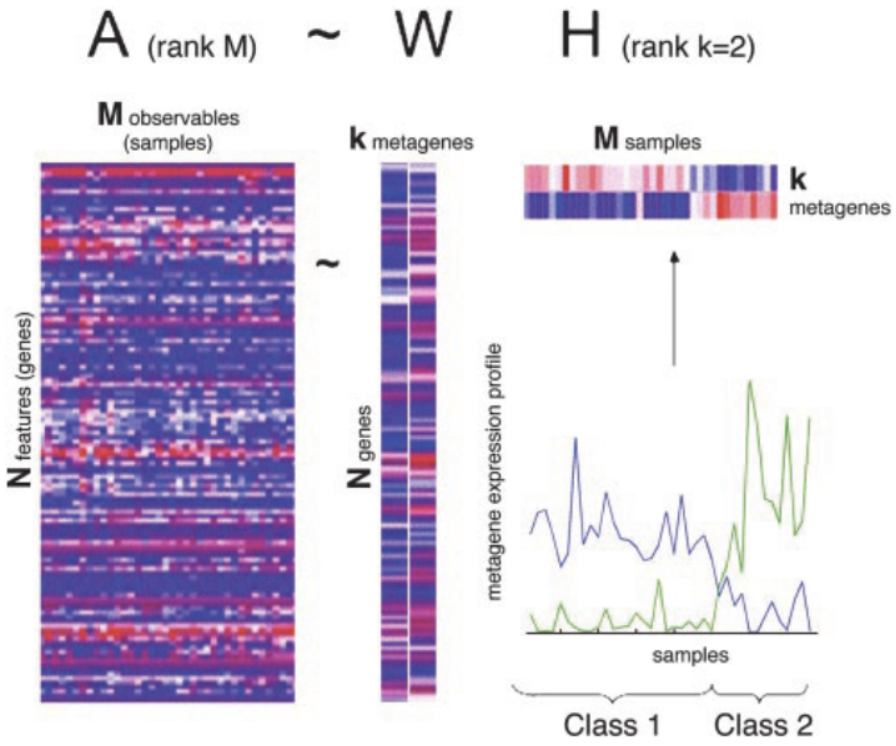


Fig. 1. A rank-2 reduction of a DNA microarray of *N* genes and *M* samples is obtained by NMF, $A \sim WH$. For better visibility, *H* and *W* are shown with exaggerated width compared with original data in *A*, and a white line

0.17 From the *NMF* vignette by Renaud Gaujoux

The main approach to NMF is to estimate matrices W and H as a local minimum:

$$\min_{W,H \geq 0} \underbrace{[D(X, WH) + R(W, H)]}_{=F(W,H)} \tag{2}$$

where

- D is a loss function that measures the quality of the approximation. Common loss functions are based on either the Frobenius distance

$$D : A, B \mapsto \frac{Tr(AB^t)}{2} = \frac{1}{2}\sum_{ij}(a_{ij} - b_{ij})^2,$$

  or the Kullback-Leibler divergence.

$$D : A, B \mapsto KL(A||B) = \sum_{i,j} a_{ij} \log \frac{a_{ij}}{b_{ij}} - a_{ij} + b_{ij}.$$

- R is an optional regularization function, defined to enforce desirable properties on matrices W and H, such as smoothness or sparsity (Cichocki et al. 2008).

0.18 Factor the matrix of expression measures

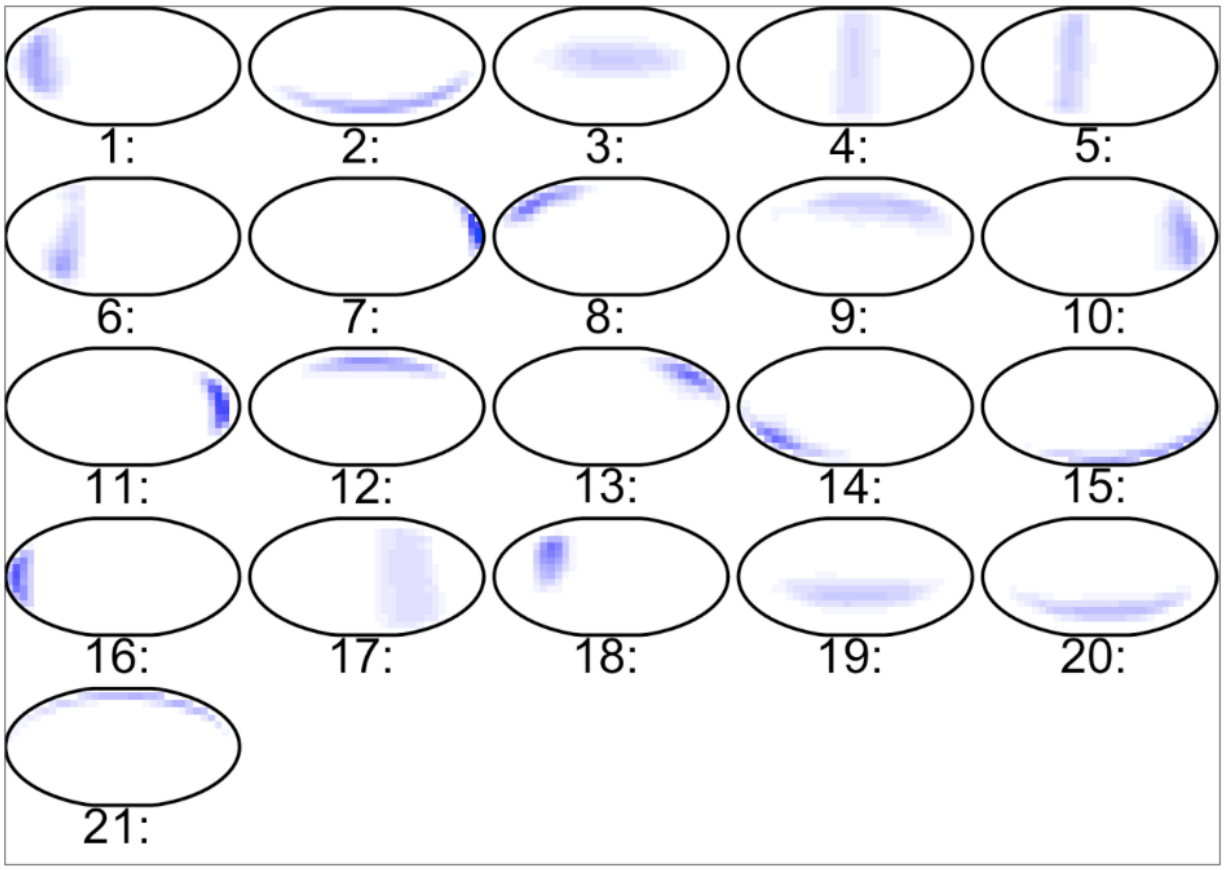
- Rows are positions in the reregistered ellipse
- Columns are genes

```
mm = nmf(uex, rank=21) # takes a minute on macbook
```

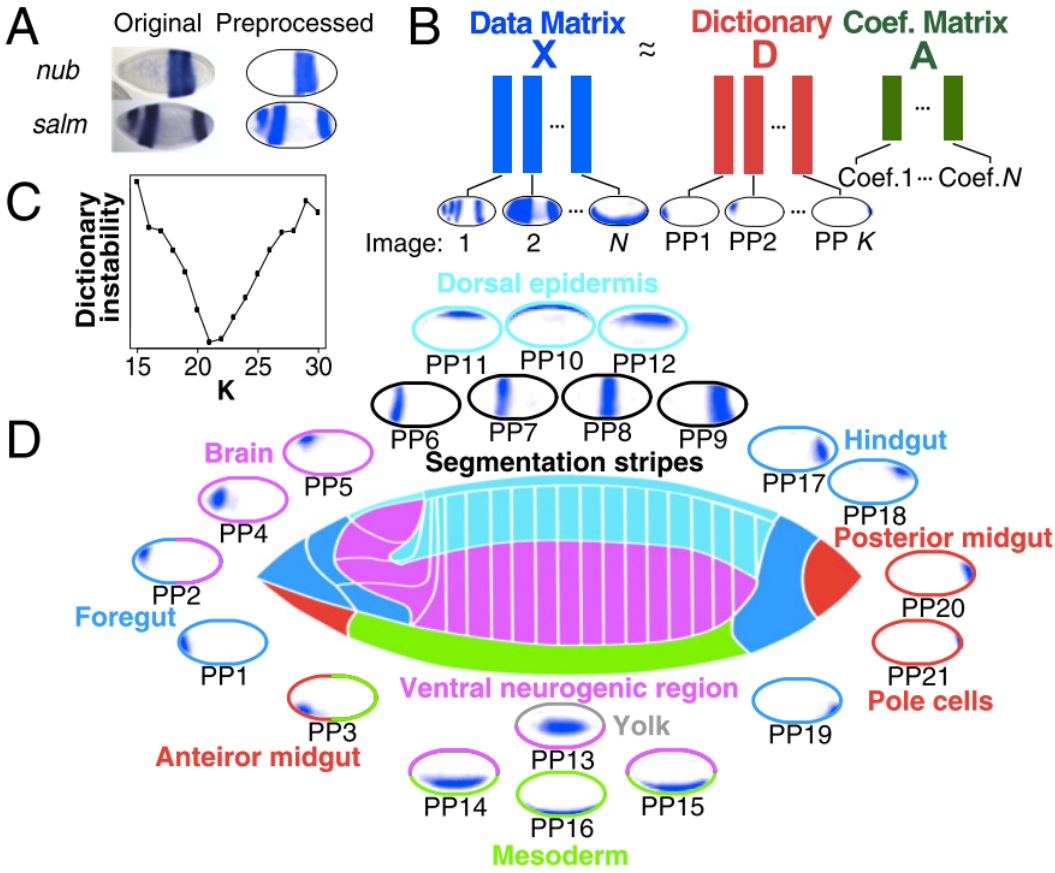
```
<Object of class: NMFfit>
 # Model:
  <Object of class:NMFstd>
  features: 405
  basis/rank: 21
  samples: 701
 # Details:
  algorithm:  brunet
  seed:  random
  RNG: 403L, 1L, ..., 1716923164L [baff3023b8693dbef07d065d4e2b4db6]
  distance metric:  'KL'
  residuals:  2766.095
  Iterations: 2000
  Timing:
     user  system elapsed
   72.953   3.245  77.848
```

0.19 Project the basis vectors to the blastocyst template

```
imageBatchDisplay(basis(mm), nrow=5, ncol=5, template=template)
```



0.20 An assignment of "principal patterns"



0.21 Comments

- *Curse of dimensionality:* as the number of features increases, utility of distance metrics for object grouping diminishes (space is mostly empty, distances generally small)
- *Bet on sparsity principle:* favor procedures that are able to prune features/dimensions, because in non-sparse case, nothing works
- All the results displayed are tunable, could be interactive
- Sensitivity analysis: Enhance the capacity of reports to demonstrate their own robustness

0.22 Remainder of talk

- Bioconductor strategies: user interface and object designs
- Cluster analysis formalities; hclustWidget
- Classifier formalities; mlearnWidget

0.23 On the user interface

- The method is primary (constituents of CRAN task view "MachineLearning")
- What does the learner consume?
  - data in a specific format, tuning parameters

- What does the learner emit?
  - an object with scores, assignments, metadata about the run
- Aims
  - reduce complexity of user tasks
  - capitalize on formal structuring of containers for inputs and outputs
  - foster sensitivity analysis
- We'll now use a modified MLInterfaces::hclustWidget that capitalizes on these notions

0.24 Exploring clusters with tissue-of-origin data

```
nicehclustWidget(t(etiss))
```

0.25 Some definitions: general distance

Definition [edit]

A **metric** on a set X is a function (called the *distance function* or simply **distance**)

$$d : X \times X \to \mathbf{R},$$

where **R** is the set of real numbers, and for all x, y, z in X, the following conditions are satisfied:

1. $d(x, y) \geq 0$ (*non-negativity*, or separation axiom)
2. $d(x, y) = 0$ if and only if $x = y$ (*identity of indiscernibles*, or coincidence axiom)
3. $d(x, y) = d(y, x)$ (*symmetry*)
4. $d(x, z) \leq d(x, y) + d(y, z)$ (*subadditivity / triangle inequality*).

Conditions 1 and 2 together define a *positive-definite function*. The first condition is implied by the others.

0.26 Examples:

0.26.1 Euclidean distance

- High-school analytic geometry: distance between two points in R^3
- $p_1 = (x_1, y_1, z_1)$, $p_2 = (x_2, y_2, z_2)$
- $\Delta x = x_1 - x_2$, etc.
- $d(p_1, p_2) = \sqrt{(\Delta x)^2 + (\Delta y)^2 + (\Delta z)^2}$

0.26.2 Manhattan distance

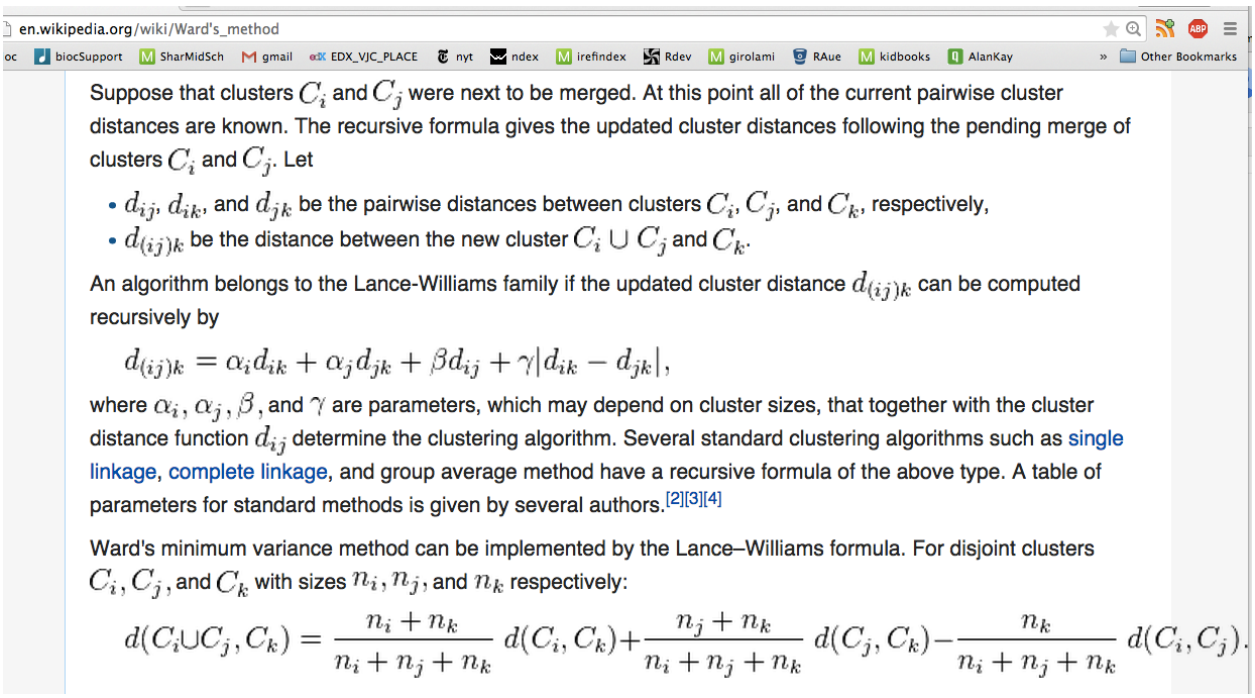
- $d(p_1, p_2) = |\Delta x| + |\Delta y| + |\Delta z|$

0.26.3 New concept of distance for categorical vectors:

Sam Buttrey and Lyn Whitaker's *treeClust* (R Journal article)

0.27 What is the ward.D2 agglomeration method?

- Enables very rapid update upon change of distance or # genes



0.28 What is the Jaccard similarity coefficient?

The **Jaccard index**, also known as the **Jaccard similarity coefficient** (originally coined *coefficient de communauté* by Paul Jaccard), is a statistic used for comparing the similarity and diversity of sample sets. The Jaccard coefficient measures similarity between finite sample sets, and is defined as the size of the intersection divided by the size of the union of the sample sets:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}.$$

(If A and B are both empty, we define $J(A,B) = 1$.)

$$0 \leq J(A,B) \leq 1.$$

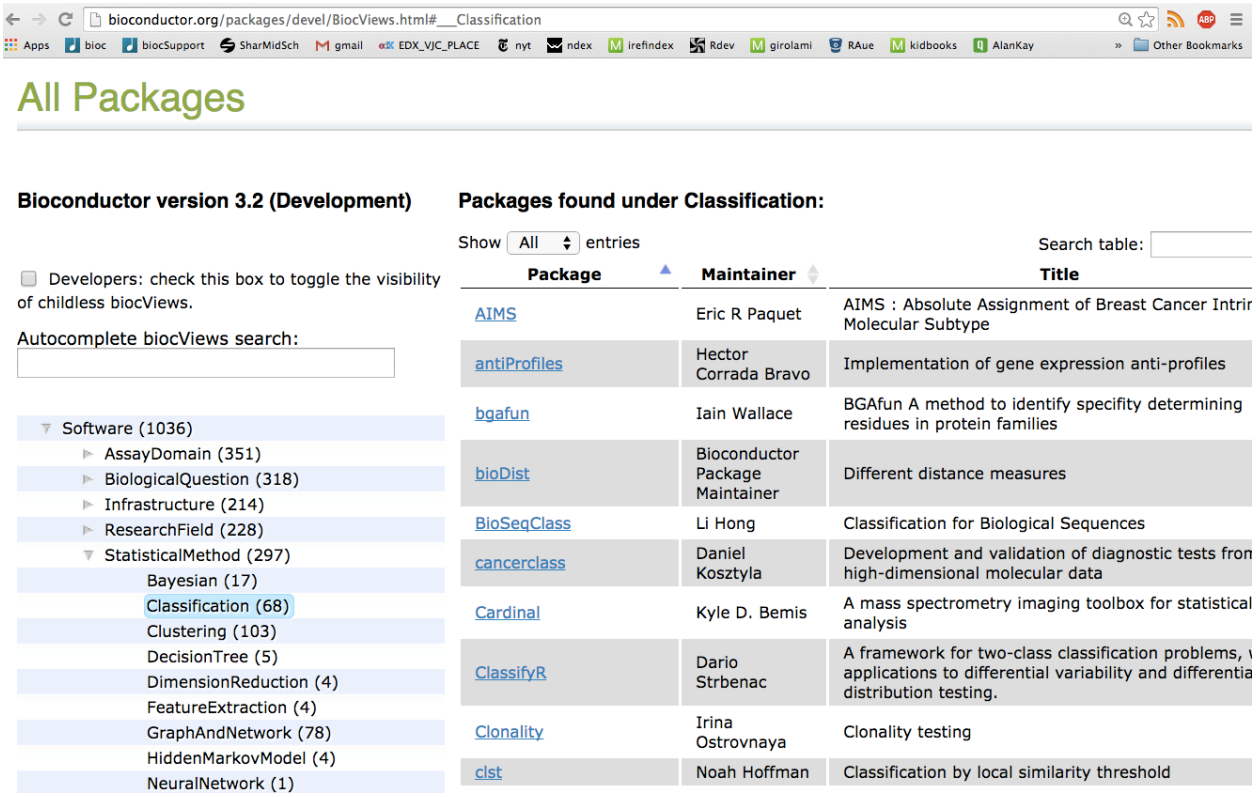
0.29 Summary

- Hierarchical clustering is tunable; distance, fusion method, feature selection all have impact
- There are other principles/algorithms: divisive, semi-supervised, model-based
- Other figures of merit: consensus, gap statistic
- See the *mlr* for structured interface

0.30 On classification methods with genomic data

- Vast topic
- Key resources in R:
  - Machine Learning task view at CRAN
  - 'metapackage' mlr
- In Bioconductor, consider
  - The 'StatisticalMethod' task view (next slide)
  - MLInterfaces (a kind of metapackage)

0.31 BiocViews: StatisticalMethod



0.32 Conceptual basis for methods covered in the talk

- "Two cultures" of statistical analysis (Leo Breiman)
  - model-based
  - algorithmic
- Ideally you will understand and use both
  - $X \sim N_p(\mu, \Sigma)$, seek and use structure in μ, Σ as estimated from data; pursue weakening of model assumptions
  - $y \approx f(x)$ with response y and features x, apply agnostic algorithms to the data to choose f and assess the quality of the prediction/classification

0.33 A method on the boundary: linear discriminant analysis

- The idea is that we can use a linear combination of features to define a score for each object
- The value of the score determines the class assignment
- This assumes that the features are quantitative and are measured consistently for all objects
- for p-dimensional feature vector x with prior probability π_k, mean μ_k for class k, and common covariance matrix for all classes

$$\delta_k(x) = x^t \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^t \Sigma^{-1} \mu_k + \log \pi_k$$

is the discriminant function; x is assigned to the class for which $\delta_k(x)$ is largest

0.34 Notes on LDA

- It is "on the boundary" because it can be justified using parametric modeling assumptions, assigning to maximize likelihood ratio
- Algorithmic arguments justify the criterion as it maximizes ratio of between- to within-class variances among all linear combinations of features (Fisher)
- Further algorithmic arguments lead to variations based on regularization concepts

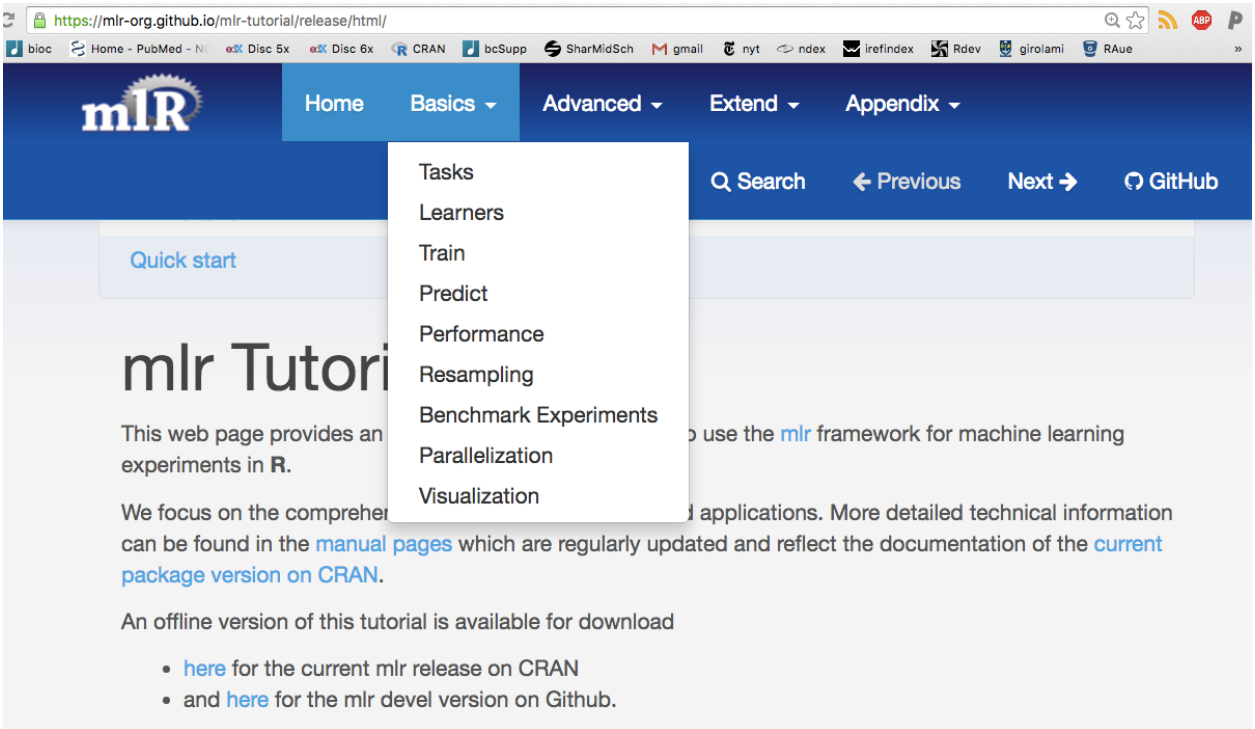
0.35 Other approaches, issues

- Direct "learning" of statistical parameters in regression or neural network models
- Recursive partitioning of classes, repeating searches through all features for optimal discrimination
- Ensemble methods in which votes are assembled among different learners or over perturbations of the data
- Unifying loss-function framework: see *Elements of statistical learning* by Hastie, Tibshirani and Friedman
- Figures of merit: misclassification rate (cross-validated), AUROC

0.36 A demonstration with tissue-of-origin expression data

```
mlearnWidget(tiss, infmla=Tissue~.)
```

0.37 check out mlr and consider how MLInterfaces could employ it



0.38 Remarks

- all examples here employ mature, reduced data
- statistical learning also important at early stages, but data volume leads to challenges
- interactive modeling/learning as the product
- in opposition to a potentially overoptimistic selection
- new work on post-selection inference in *selectiveInference*